```
// Standard API
document.createElement('div');

// Namespaced element
document.createElementNS('http://www.w3.org/1999/xhtml', 'div');

// HTML parsing APIs
(new DOMParser()).parseFromString('<div></div>', 'text/html').body.firstChild;

// Range API
document.createRange().createContextualFragment('<div></div>').firstChild;

// Template element
Object.assign(document.createElement('template'), {innerHTML: '<div>
</div>'}).content.firstChild;

// iframe with srcdoc (creates a new DOM tree)
let iframe = document.createElement('iframe'); iframe.srcdoc = '<div></div>'; //
access via iframe.contentDocument.body.firstChild

// XSLTProcessor (rare, but possible)
let xslt = (new DOMParser()).parseFromString(
  `<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
     <xsl:template match="/"><div xmlns="http://www.w3.org/1999/xhtml"/>
</xsl:template>
   </xsl:stylesheet>`, 'application/xml');
let processor = new XSLTProcessor(); processor.importStylesheet(xslt);
processor.transformToFragment(document.implementation.createDocument('', '',
null), document);

// document.implementation.createDocument (XML, but can create elements)
document.implementation.createDocument('http://www.w3.org/1999/xhtml',
'div').documentElement;

// document.implementation.createHTMLDocument
document.implementation.createHTMLDocument().createElement('div');

// Custom Elements (constructor, if not banned)
class MyDiv extends HTMLDivElement {}; customElements.define('my-div', MyDiv,
{extends: 'div'}); new MyDiv();

// Using importNode (cloning from another document)
let doc = document.implementation.createHTMLDocument(); let div =
doc.createElement('div'); document.importNode(div, true);

// Using cloneNode on a template
document.createElement('template').content.cloneNode(true).appendChild(document.cr
eateElement('div'));
```

Finding existing DOM nodes

```
// Standard selectors
document.getElementById('myDiv');
document.getElementsByClassName('myClass')[0];
document.getElementsByTagName('div')[0];
document.getElementsByName('myDivName')[0];
document.querySelector('div');
document.querySelectorAll('div')[0];

// Traversal APIs
document.body.firstChild;
document.body.lastChild;
document.body.childNodes[0];
document.body.children[0];
document.body.parentNode;
document.body.nextSibling;
document.body.previousSibling;
document.body.nextElementSibling;
document.body.previousElementSibling;

// TreeWalker
document.createTreeWalker(document.body, NodeFilter.SHOW_ELEMENT).nextNode();

// NodeIterator
document.createNodeIterator(document.body, NodeFilter.SHOW_ELEMENT).nextNode();

// Range API
let range = document.createRange(); range.selectNode(document.body);
range.startContainer;

// Event target (from an event handler)
document.body.addEventListener('click', e => { let node = e.target; });

// Shadow DOM
let shadow = document.body.attachShadow({mode: 'open'}); shadow.host;

// Selection API
window.getSelection().anchorNode;

// Active element
document.activeElement;

// OwnerDocument
document.body.ownerDocument;

// ParentNode API
document.body.closest('html');

// Custom Elements (via lifecycle callbacks)
customElements.define('my-div', class extends HTMLElement {
  connectedCallback() { let node = this; }
});

// HTMLCollection/NodeList iteration
for (let node of document.body.children) { /* node is a reference */ }

// iframe content
```

```
let iframe = document.querySelector('iframe'); iframe.contentDocument.body;

// window.frameElement
window.frameElement;

// document.currentScript
document.currentScript;

// document.scripts
document.scripts[0];

// document.forms, document.images, document.links, document.anchors
document.forms[0];
document.images[0];
document.links[0];
document.anchors[0];

// document.head, document.body, document.documentElement
document.head;
document.body;
document.documentElement;

// document.defaultView (returns window, but can access DOM via window.document)
document.defaultView.document.body;

// document.adoptNode (returns a reference to the adopted node)
let node = document.createElement('div'); document.adoptNode(node);

// document.importNode (returns a reference to the imported node)
let node2 = document.createElement('div'); document.importNode(node2);

// document.elementFromPoint
document.elementFromPoint(0, 0);

// document.elementsFromPoint
document.elementsFromPoint(0, 0)[0];

// document.scrollingElement
document.scrollingElement;

// document.pointerLockElement
document.pointerLockElement;

// document.fullscreenElement
document.fullscreenElement;

// document.getRootNode
document.body.getRootNode();

// document.getSelection
window.getSelection().focusNode;

// MutationObserver (callback receives references to added/removed nodes)
new MutationObserver((mutations) => { mutations.forEach(m => m.addedNodes[0]);
}).observe(document.body, {childList: true});
```

```
// CSSStyleSheet.ownerNode
document.styleSheets[0].ownerNode;

// HTMLTableElement.rows/cells
document.querySelector('table').rows[0].cells[0];

// HTMLFormElement.elements
document.querySelector('form').elements[0];

// HTMLSelectElement.options
document.querySelector('select').options[0];

// HTMLMapElement.areas
document.querySelector('map').areas[0];

// HTMLMediaElement.textTracks/cues
document.querySelector('video').textTracks[0].cues[0];

// SVG-specific: getElementById, getElementsByTagName, etc. on SVGSVGElement
document.querySelector('svg').getElementById('svgDiv');

// XPath
document.evaluate('//div', document, null, XPathResult.FIRST_ORDERED_NODE_TYPE,
null).singleNodeValue;
```

Globally Available DOM API Types

Core Node Types

- Node
- Element
- Document
- DocumentFragment
- Text
- Comment
- CDATASection // XML documents only
- ProcessingInstruction
- DocumentType
- Attr

Element Subclasses

HTML Elements (all inherit from `HTMLElement`)

- HTMLElement
- HTMLAnchorElement
- HTMLAreaElement
- HTMLAudioElement

- HTMLBaseElement
- HTMLBodyElement
- HTMLBRElement
- HTMLButtonElement
- HTMLCanvasElement
- HTMLDataElement
- HTMLDataListElement
- HTMLDetailsElement
- HTMLDialogElement
- HTMLDivElement
- HTMLDListElement
- HTMLEmbedElement
- HTMLFieldSetElement
- HTMLFormElement
- HTMLHeadingElement
- HTMLHeadElement
- HTMLHRElement
- HTMLHtmlElement
- HTMLIFrameElement
- HTMLImageElement
- HTMLInputElement
- HTMLLabelElement
- HTMLLegendElement
- HTMLLIElement
- HTMLLinkElement
- HTMLMapElement
- HTMLMetaElement
- HTMLMeterElement
- HTMLModElement
- HTMLOListElement
- HTMLObjectElement
- HTMLOptGroupElement
- HTMLOptionElement
- HTMLOutputElement
- HTMLParagraphElement
- HTMLParamElement
- HTMLPictureElement
- HTMLPreElement
- HTMLProgressElement
- HTMLQuoteElement
- HTMLScriptElement
- HTMLSelectElement
- HTMLSlotElement
- HTMLSourceElement
- HTMLSpanElement

- HTMLStyleElement
- HTMLTableCaptionElement
- HTMLTableCellElement
- HTMLTableColElement
- HTMLTableElement
- HTMLTimeElement
- HTMLTitleElement
- HTMLTableRowElement
- HTMLTableSectionElement
- HTMLTemplateElement
- HTMLTextAreaElement
- HTMLTrackElement
- HTMLUListElement
- HTMLUnknownElement
- HTMLVideoElement

SVG Elements (all inherit from `SVGElement`)

- SVGElement
- SVGSVGElement
- SVGCircleElement
- SVGClipPathElement
- SVGDefsElement
- SVGEllipseElement
- SVGFEBlendElement
- SVGFEColorMatrixElement
- SVGFEComponentTransferElement
- SVGFECompositeElement
- SVGFEConvolveMatrixElement
- SVGFEDiffuseLightingElement
- SVGFEDisplacementMapElement
- SVGFEDistantLightElement
- SVGFEDropShadowElement
- SVGFEFloodElement
- SVGFEFuncAElement
- SVGFEFuncBElement
- SVGFEFuncGElement
- SVGFEFuncRElement
- SVGFEGaussianBlurElement
- SVGFEImageElement
- SVGFEMergeElement
- SVGFEMergeNodeElement
- SVGFEMorphologyElement
- SVGFEOffsetElement
- SVGFEPointLightElement

- SVGFESpecularLightingElement
- SVGFESpotLightElement
- SVGFETileElement
- SVGFETurbulenceElement
- SVGFilterElement
- SVGForeignObjectElement
- SVGGElement
- SVGImageElement
- SVGLineElement
- SVGLinearGradientElement
- SVGMarkerElement
- SVGMaskElement
- SVGPathElement
- SVGPatternElement
- SVGPolygonElement
- SVGPolylineElement
- SVGRadialGradientElement
- SVGRectElement
- SVGStopElement
- SVGSwitchElement
- SVGSymbolElement
- SVGTSpanElement
- SVGTextElement
- SVGTextPathElement
- SVGTitleElement
- SVGUseElement
- SVGViewElement

MathML Elements

MathMLElement

Document Types

- HTMLDocument
- XMLDocument
- SVGDocument

Other DOM-Related Classes

- CharacterData
- Range
- Selection
- MutationObserver
- MutationRecord
- TreeWalker

- NodeIterator
- NamedNodeMap
- DOMTokenList
- DOMStringMap
- DOMRect
- DOMRectList
- DOMPoint
- DOMPointReadOnly
- DOMMatrix
- DOMMatrixReadOnly
- DOMParser
- XMLSerializer
- XPathEvaluator
- XPathResult
- XPathExpression

NodeList/Collections

- NodeList
- HTMLCollection
- HTMLAllCollection
- HTMLFormControlsCollection
- HTMLOptionsCollection
- SVGLengthList
- SVGNumberList
- SVGPointList
- SVGStringList
- SVGTransformList

Attribute/Style/Other

- Attr
- NamedNodeMap
- CSSStyleDeclaration
- StyleSheet
- CSSStyleSheet
- CSSRuleList
- CSSRule

Miscellaneous

- Comment
- Text
- ProcessingInstruction
- CDATASection
- DocumentType

- ShadowRoot
- DocumentFragment

Event Classes

(tend to have references to DOM)

- Event
- CustomEvent
- UIEvent
- MouseEvent
- KeyboardEvent
- FocusEvent
- InputEvent
- TouchEvent
- PointerEvent
- WheelEvent
- CompositionEvent
- DragEvent
- ClipboardEvent
- AnimationEvent
- TransitionEvent
- PopStateEvent
- HashChangeEvent
- PageTransitionEvent
- ProgressEvent
- StorageEvent
- MessageEvent
- ErrorEvent
- PromiseRejectionEvent

Same but with inheritance depicted

```
graph TD

  %% Inheritance
  EventTarget -->|inherits| Node
  EventTarget -->|inherits| Window
  EventTarget -->|inherits| XMLHttpRequest
  EventTarget -->|inherits| EventSource
  EventTarget -->|inherits| WebSocket
  EventTarget -->|inherits| Worker
  EventTarget -->|inherits| MessagePort
  EventTarget -->|inherits| BroadcastChannel
  EventTarget -->|inherits| FileReader
  EventTarget -->|inherits| MediaStream
  EventTarget -->|inherits| MediaRecorder
  EventTarget -->|inherits| RTCPeerConnection
  EventTarget -->|inherits| Event

  Node -->|inherits| Document
  Node -->|inherits| DocumentFragment
  Node -->|inherits| Element
  Node -->|inherits| Text
  Node -->|inherits| Comment
  Node -->|inherits| CDATASection
  Node -->|inherits| ProcessingInstruction
  Node -->|inherits| DocumentType
  Node -->|inherits| Attr

  CharacterData -->|inherits| Text
  CharacterData -->|inherits| Comment
  CharacterData -->|inherits| CDATASection

  Document -->|inherits| XMLDocument
  Document -->|inherits| HTMLDocument
  Document -->|inherits| SVGDocument

  Element -->|inherits| SVGElement
  Element -->|inherits| MathMLElement
  Element -->|inherits| HTMLElement

  DocumentFragment -->|inherits| ShadowRoot

  %% Aggregation/Fields

  NodeList -- "item[]" --> Node
  HTMLCollection -- "item[]" --> Element
  NamedNodeMap -- "item[]" --> Attr
  DOMTokenList -- "item[]" --> string
  DOMStringMap -- "item[]" --> string

  Range -- "startContainer" --> Node
  Range -- "endContainer" --> Node

  Selection -- "anchorNode" --> Node
  Selection -- "focusNode" --> Node

  MutationRecord -- "target" --> Node
```

```
MutationRecord -- "addedNodes" --> NodeList
MutationRecord -- "removedNodes" --> NodeList

TreeWalker -- "root" --> Node
TreeWalker -- "currentNode" --> Node

NodeIterator -- "root" --> Node
NodeIterator -- "referenceNode" --> Node

ShadowRoot -- "host" --> Element

Attr -- "ownerElement" --> Element

Document -- "documentElement" --> Element
Document -- "body" --> Element
Document -- "head" --> Element

Element -- "attributes" --> NamedNodeMap
Element -- "children" --> HTMLCollection
Element -- "classList" --> DOMTokenList
Element -- "dataset" --> DOMStringMap

DocumentFragment -- "childNodes" --> NodeList

%% Miscellaneous
DOMRectList -- "item[]" --> DOMRect
CSSRuleList -- "item[]" --> CSSRule
StyleSheet -- "ownerNode" --> Node
CSSStyleSheet -- "ownerNode" --> Node

%% Standalone classes (not directly connected)
DOMRect
DOMPoint
DOMPointReadOnly
DOMMatrix
DOMMatrixReadOnly
DOMParser
XMLSerializer
XPathEvaluator
XPathResult
XPathExpression
```

Top comments (0)

Subscribe